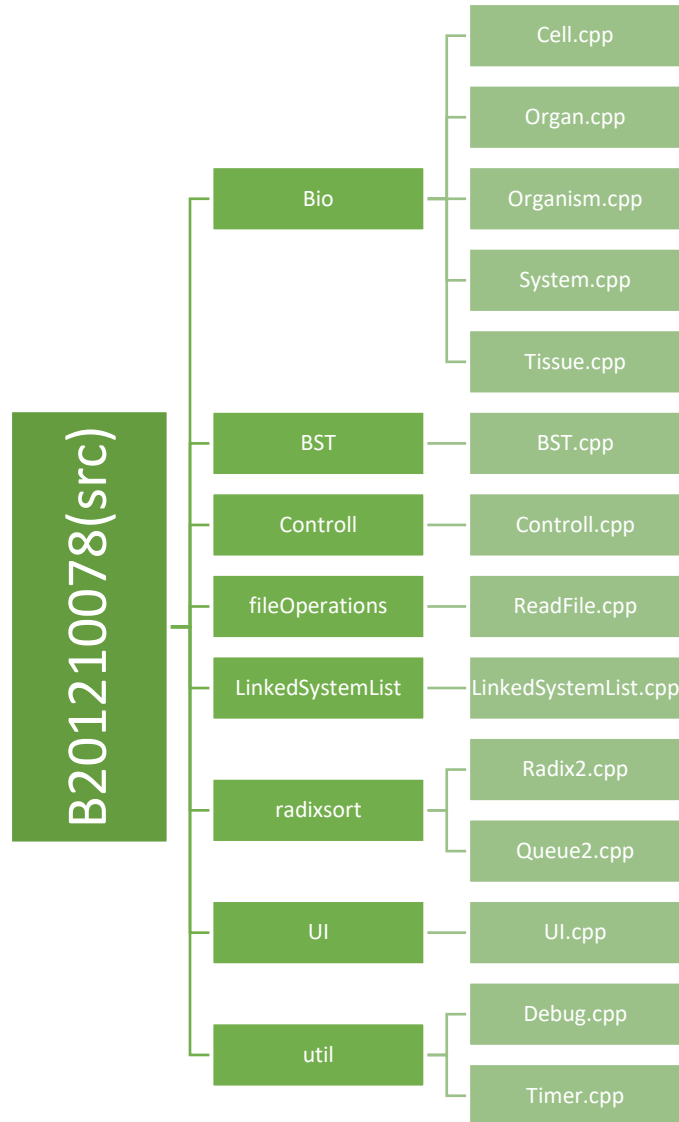
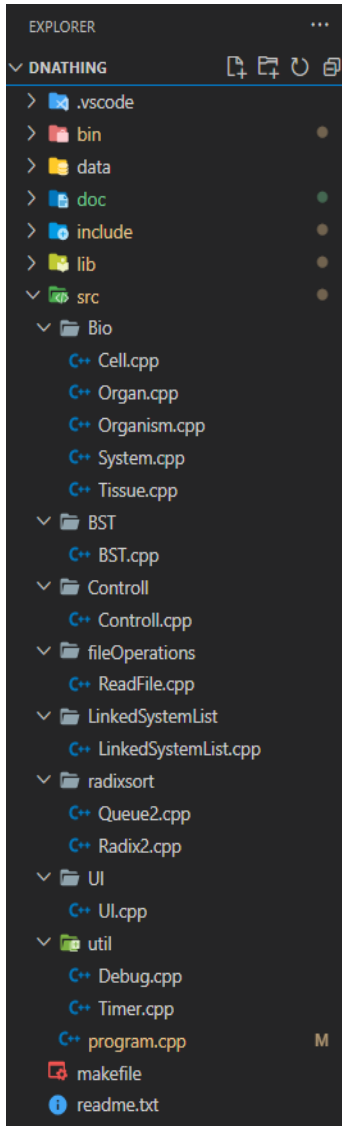


## Organizma Oluşturma Uygulaması

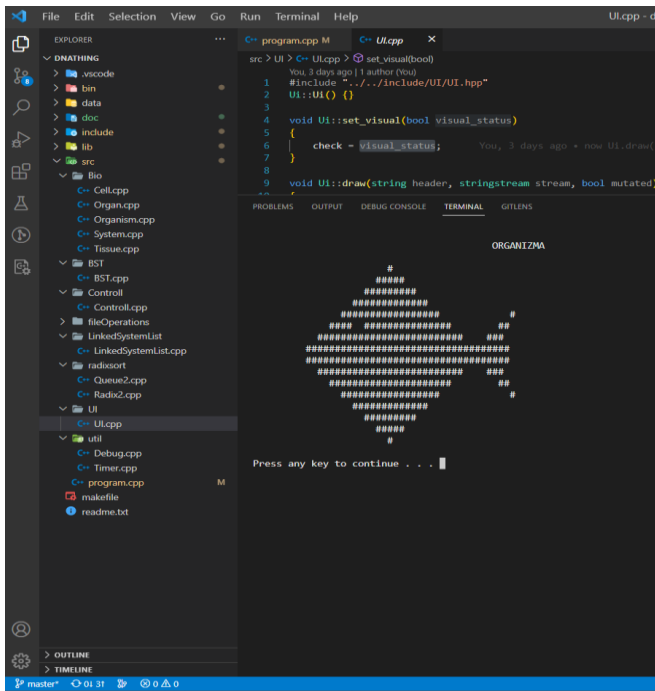


Ödevime başlamadan önce henüz tasarım aşamasındayken ödevin modüler yapıda olmasına karar verdim ve ödevi parçalara böldüm. **Src** üzerinden aktarmak icap ederse: Senaryodaki biyolojik varlıkların hepsini “**Bio**” dosyasında, gerekli veri yapılarını “**BST**”, “**radixsort**” ve “**LinkedSystemList**” klasörlerinde tuttum. “**fileOperations**” klasöründe bulunan “**ReadFile.cpp**” kaynak dosyasında “**data.txt**” adlı “**data**” klasöründeki txt dosyasını okudum. Okunan verileri “**Controll**” klasöründeki “**Controll.cpp**” dosyasına gönderdim ve işleyerek bilgi haline getirdim. “**Controll.cpp**” de “**data.txt**” deki tüm sayılar bir canlı oluşturur gibi önce hücrelere sonra dokulara ve dokular organlara, organlar sistemlere son olarak sistemler ise organizmayı oluşturdu. Mutasyon işlemini ise her biyolojik yapının içinde bir zincir şeklinde organizmadan hücreye olacak şekilde yaptım. Harika kod yazdığımı düşünmüyorum ancak hata yaparak doğruyu bulmaya çalışıyorum bu ödevde başlamadan önce C++ dilinde nasıl daha hızlı string-integer dönüşümü yaparım konusunu araştırdım ve bazı metotları bulup denedim. Bu metotların hız testlerini “**util**” klasöründeki “**Timer.cpp**” dosyasındaki zaman metotları ile ölçtüm. Ödevdeki örnek verilerin okunması bahsinde kayda

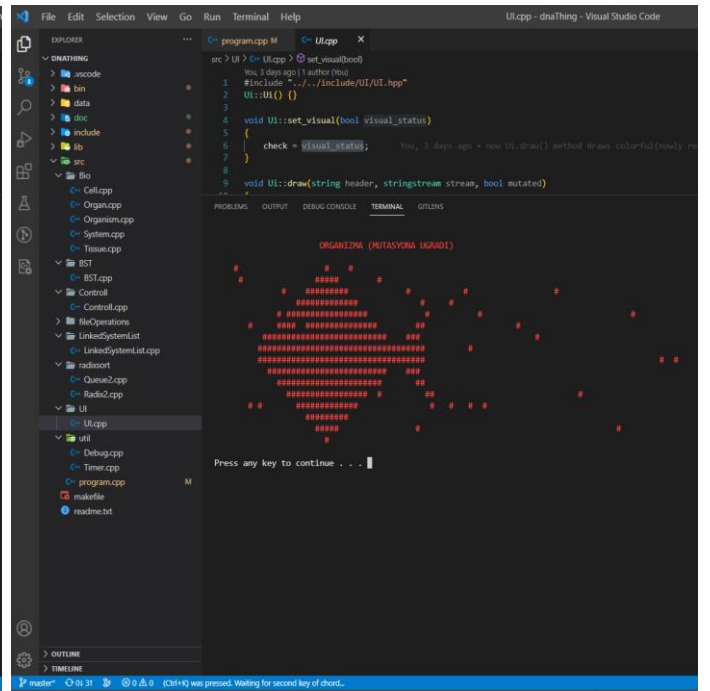
Değer bir hız artışı yaşamadım.” **util**” klasöründeki bir diğer dosya olan *”Debug.cpp”* de programın ‘debug’ modunda çalışıp çalışmayacağıyla ilgili ve ‘debug’ moduna özel metotlar yazdım. Ödevi yaparken çokça test amaçlı ‘cout<<test’ yazmaktansa görünürlüğü mainin içinden belirlenebilen ‘changeDebugStatus(1)’ ile ‘debug’ test metotları yazdım. Kodun kalitesini arttırdığını düşünüyorum. Size ‘debugMode=0’ halinde göndereceğim ödevi ancak programın çalışırken geçirdiği zaman gibi basit bilgileri görmek isterseniz mainin içindeki ‘Debug::change\_debug\_status(0)’ satırının parametresini ‘1’ ile değiştirmeniz yeterli olacaktır.

Organizmamın çıktısını-görüntüsünü ‘Controll’ sınıfında ‘showCreature()’ isimli metodun -stringstream- çıktısı olarak elde ettim. Bu çıktıyı ise **”UI”** klasöründe *”UI.cpp”* dosyasında ‘draw()’ metodu ile konsola çıktı olarak bastırdım. Dengesiz ağaçların ödev anlatımında ‘#’ karakteri ile basıldığı gösterilmiş derseniz ‘Controll’ sınıfında ‘showCreature()’ metoduna parametre olarak gireceğiniz karakter varsayılan ‘#’ ile değişebilmektedir. İlaveten ödevde renk katmak istedim ve ‘UI’ sınıfında mutasyonlu çıktıyı kırmızı renkte olacak şekilde konsola bastırdım. Bunu yapabilmek için ‘windows.h’ kütüphanesinin ‘setConsoleTextAttribute()’ metodunu kullandım.

Ödevimi iki farklı bilgisayarda derledim ikisinde de başarılı çıktılar aldım



Şekil 2 Organizma



Şekil 1 Organizma (Mutasyona uğramış)

**Mustafa BİÇER, B201210078, 1A**

Son ödev teslim tarihinden sonra ödevi [githubımda](#) public hale getireceğim.